



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,144	10/31/2003	Dhruva Ranjan Chakrabarti	200313003-1	3438
22879 7590 12/17/2010 HEWLETT-PACKARD COMPANY Intellectual Property Administration 3404 E. Harmony Road Mail Stop 35 FORT COLLINS, CO 80528				
EXAMINER WU, JUNCHUN				
ART UNIT 2191		PAPER NUMBER		
NOTIFICATION DATE 12/17/2010		DELIVERY MODE ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
laura.m.clark@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte DHURVA RANJAN CHAKRABARTI and SHIN-MING LIU

Appeal 2009-008812
Application 10/699,144
Technology Center 2100

Before LANCE LEONARD BARRY, JAY P. LUCAS, and
CAROLYN D. THOMAS, *Administrative Patent Judges*.

THOMAS, *Administrative Patent Judge*.

DECISION ON APPEAL¹

¹ The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the “MAIL DATE” (paper delivery mode) or the “NOTIFICATION DATE” (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from a final rejection of claims 1, 3-8, and 10-21. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

According to Appellants, the invention relates to software compilers (*see* Spec. 1:11-12). More specifically, Appellants' invention involves (1) receiving a plurality of modules of source code; (2) generating intermediate representations corresponding to the modules; and (3) extracting a set of data from the intermediate representations to create an inliner summary for each module. The inliner summaries and a globally-sorted working list order are analyzed to determine which call sites in the modules are to be inlined by substituting code from a called module. (Spec., Abstract).

Claim 1 is illustrative:

1. A method of compiling a computer program, the method comprising:

receiving a plurality of modules of source code;

generating intermediate representations corresponding to the modules;

extracting a set of data from the intermediate representations to create an inliner summary for each module;

using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase, without using the intermediate representations in the inline analysis phase, to determine which call sites in the modules are to be inlined by substituting code from a

called module, wherein said globally-sorted working-list based order is dynamically updated during the inline analysis phase; and

after a call site is determined to be inlined, updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph,

wherein determining the call sites to be inlined involves proceeding only once through the call sites in said dynamically-updated globally-sorted working-list based order.

Rejection

Claims 1, 3-8, and 10-21 are rejected under 35 U.S.C. § 103 (a) as being unpatentable over Ayers et al. (ANDREWS AYERS ET AL., AGGRESSIVE INLINING, (1997) and Schmidt (US Patent No. 6,195,793, B1, Feb. 27, 2001).

FINDINGS OF FACT (FF)

Schmidt Reference

1a. Schmidt discloses: “In FIGS. 2 and 3, sequential steps of the TABLE 1 algorithm for initialization, first approximation of inline candidates, and refinement of the first approximation of inline candidates are shown.” (col. 5, ll. 18-21).

1b. Schmidt discloses: “With inlining, a call to a procedure can be replaced by a duplicate of the body of the called procedure” (col. 1, ll. 20-21). Schmidt also discloses that “an arc represents a call from the

originating node's procedure to the target node's procedure" (col. 1, ll. 52-53).

1c. Schmidt discloses:

In the first approximation phase of FIG. 2, all inline candidates or arcs with finite priority are placed in the initial priority queue. Then as indicated at a block 306, the best inline candidates are removed from the initial priority queue, marking them as tentatively to be inlined, until the code bloat budget has been exhausted. This initial queue is then discarded. At this point the best sites to be inlined have been determined, provided that none of the inlined procedures grow because they themselves contain inlined call sites. (*see* col. 5, ll. 59-67).

1d. Schmidt discloses: "In the refinement stage, the compilation order for procedures is determined and the determination of which procedures to inline possibly is changed, based on changes in procedure size." (col. 6, lines 1-4).

1e. Schmidt discloses: Two more priority queues are used to accomplish this, *i.e.*, a ReadyQueue and an AuxQueue.

A ReadyQueue contains all procedures that can be issued next in the compilation order, either because they are leaf procedures The AuxQueue contains initially outlined call sites whose final bloat is known, and that may be used to replace those candidates that have bloated unacceptably since they were initially selected. (*see* col. 6, ll. 5-20).

1f. Schmidt discloses:

All leaf procedures are placed in the ready queue as indicated at a block 310. Then the best procedure is ... selected from the ready queue and processed First the best procedure is appended to the compilation order list as indicated at a block 316. The bloat of the best procedure is calculated as indicated Any subsequent inlining of call sites within the procedure itself will cause its bloat to increase. Then all outgoing arcs are processed as indicated at a block 320 to confirm or reject the initial inlining decisions made about the outgoing arcs (see col. 6, ll. 21-31).

PRINCIPLES OF LAW

Obviousness

In rejecting claims under 35 U.S.C. § 103, it is incumbent upon the Examiner to establish a factual basis to support the legal conclusion of obviousness. *See In re Fine*, 837 F.2d 1071, 1073 (Fed. Cir. 1988). If the Examiner's burden is met, the burden then shifts to the Appellants to overcome the prima facie case with argument and/or evidence. Obviousness is then determined on the basis of the evidence as a whole and the relative persuasiveness of the arguments. *See In re Oetiker*, 977 F.2d 1443, 1445 (Fed. Cir. 1992).

ANALYSIS

§103 Rejection

Claims 1, 3-8 and 10-21

Issue: Did the Examiner err in finding the prior art teaches or suggests, “determining the call sites to be inlined involves proceeding only once through the call sites ...,” as set forth in claim 1?

Appellants argue that Schmidt fails to teach or suggest “proceeding only once through the working list of call sites” (App. Br. 9) (emphasis omitted). Appellants contend that Schmidt proceeds through the call sites in more than one pass, as Schmidt discloses (1) an initial pass that occurs during the approximation phase and (2) a subsequent pass that occurs during the refinement stage (*see* App. Br. 9-10).

The Examiner finds that Schmidt discloses “determining the call sites to be inlined involves proceeding only once through the call sites in [a] working list” (Ans. 16). The Examiner acknowledges that Schmidt discloses two phases, *i.e.*, an approximation phase and a refinement phase (Ans. 16). The Examiner further finds that “the claims are open-ended with the use of the term ‘comprising,’” and that “[t]he approximation phase is separate from the refinement stage” (Ans. 16).

Here, the Examiner only looks at Schmidt’s “refinement stage” when considering whether Schmidt’s process proceeds only once through the call sites. (Ans. 16). However, we disagree with the Examiner, as we find that the Examiner has erred in relying solely on Schmidt’s “refinement stage.”

For example, Schmidt discloses that the determination of the inline candidates or arcs (*i.e.*, calls to procedures), which will be inlined, results from both the (1) approximation phase and the (2) refinement stage (*see* FF 1c-1f). For instance, Schmidt teaches that it is the “first approximation of inline candidates” that is refined (*see* FF 1a). In other words, the refinement phase is dependent on the approximation phase, as the refinement phase needs the “first approximation of inline candidates” generated by the approximation phase, before the refinement stage can begin refining.

Schmidt discloses that in the approximation phase “all inline candidates or arcs with finite priority are placed in the initial priority queue” and only “the best inline candidates are removed from the initial priority queue” (FF 1c). Further, Schmidt discloses that these “best inline candidates” are marked as “tentatively” to be inlined (FF 1c). Thus, a final determination has not been made as to whether the “best inline candidates” will be actually inlined. It is not until Schmidt’s refinement stage that “the determination of which procedures to inline possibly is changed, based on changes in procedure size” (FF 1d). In other words, inline candidates/arcs correspond to “calls to procedures” (FF 1b). When an inline candidate/arc is inlined, the actual procedure, which is called, is substituted for the inline candidate/arc (FF 1b). If the size of a procedure has changed, rendering the procedure unacceptable to be inlined, the inline candidate/arc that corresponds to the “call to that procedure” will also be unacceptable. Thus, in the refinement stage Schmidt determines, via a process that involves two priority queues (*i.e.*, a ReadyQueue and an AuxQueue) (1) which candidates (corresponding to procedures) have bloated (*i.e.*, changed to an unacceptable

size) since they were initially selected (FF 1e) and (2) confirms or rejects the initial inlining decisions (FF 1f).

Therefore, we find that while Schmidt may go through the refinement phase once to determine which of the “best inline candidates/arcs” to actually inline, the refinement stage, however, relies on the approximation phase for generating the list of *tentative best inline candidates* that it will need to refine. Accordingly, we find it was improper for the Examiner to rely solely on the refinement phase, as the refinement stage would not have happened but for the occurrence of the approximation phase.

Thus, we find that the Examiner has erred in finding that the prior art, specifically Schmidt, teaches or suggests, “determining the call sites to be inlined involves proceeding only once through the call sites ...,” as set forth in claim 1. Independent claims 8, 15, and 16 are commensurate in scope with the above-noted limitation.

Since we agree with at least one of the arguments advanced by Appellants, we need not reach the merits of Appellants’ other arguments. It follows that Appellants have shown that the Examiner erred in finding that Ayers and Schmidt renders claims 1, 3-8, and 10-21 unpatentable.

Accordingly, we reverse the Examiner’s § 103(a) rejection of independent claims 1, 8, 15, and 16, and claims 3-7, 10-14, and 17-21 which stand therewith.

DECISION

The Examiner’s rejection of claims 1, 3-8, and 10-21 under 35 U.S.C. § 103 (a) is reversed.

REVERSED

ke

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528